IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF MASSACHUSETTS

SINGULAR COMPUTING LLC, Plaintiff, v. GOOGLE LLC, Defendant.	Case No. C.A. No. 1:19-cv-12551 FDS

GOOGLE LLC'S MEMORANDUM OF LAW
IN SUPPORT OF RULE 12(B)(6) MOTION TO DISMISS
FOR LACK OF PATENTABLE SUBJECT MATTER

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

I. INTRODUCTION

The patents in suit[1] claim a processor or other device for performing what is called "low-precision, high-dynamic range" arithmetic. But Singular didn't invent low-precision arithmetic, which has existed in human endeavor as long as there has been arithmetic and has existed in computing as long as the field has existed. Nor did Singular invent high dynamic range, which is nothing more than a wide span of numbers, as the specification concedes. Nonetheless, the patents in suit attempt to monopolize a simple, abstract idea: when doing math, sometimes the exactly right calculation is unnecessary because close enough suffices. This idea is a part of daily life, whether it's deciding how much to tip at a restaurant or how to split a taxicab fare. The patents in suit claim this idea in the computing realm; however, they don't purport to improve upon existing computing technology.

The very nature of how computers perform calculations always involves some degree of imprecision. Computers must represent numbers with a finite number of bits, yet some numbers require more precision or have infinite digits—the concept of *pi* being a common example. So, when computers truncate numbers to meet their number format, this gives rise to some degree of imprecision relative to a calculation that would otherwise involve more digits than the computer's format accommodates.

Claiming this abstract idea on a low-precision, high dynamic range ("LPHDR") processor neither makes it less abstract nor adds an inventive concept. In a computer, a number is represented by a string of bits of a particular "width" (*e.g.*, 8-bit, 16-bit, 32-bit, or 64-bit). Because numbers are typically represented in a "floating point" format that follows a scientific notation format (*e.g.*, 1.25×10^{1}), the bits are used to express both precision (the left side, *i.e.*,

[1] U.S. Patent Nos. 8,407,273, 9,218,156, and 10,416,961. *See* Compl. ¶ 27.

1.25) and range (the exponent on the right side). A floating-point format can provide more precision by using more bits on the left side or provide more range by using more bits for the exponent. Thus, for a fixed processing bit-width (*i.e.*, 16-bit, 32-bit, and 64-bit), the greater the precision, the less the dynamic range (and vice versa). A LPHDR processor is nothing more than the application of the patent's abstract idea to the context of computing. By assigning more data bits to range than to precision, a LPHDR processor has less precision, but more dynamic range.

The patents in suit do not disclose a new (or indeed any) way to store numbers in a low-precision format, or with a high dynamic range; instead, they rely on the skilled artisan's knowledge of conventional approaches to LPHDR representations. The purported inventive concept appears, at most, to be the recognition that in some computing applications, some level of imprecision may be acceptable. But the specification admits that prior implementations recognized that applications could accept various levels of imprecision, and that not every application required the same level of precision. The claims therefore offer no improvement of computing technology nor any inventive concept that would render them patentable.

II. BACKGROUND

The patents in suit are directed to a "processor or other device" that "includes processing elements designed to perform arithmetic operations." '273 Patent, Abstract. The "Summary" states that in "some embodiments" of the invention, "'low precision' processing elements perform arithmetic operations which produce results that frequently differ from exact results by at least 0.1% . . . ." '273 Patent at 2:28-31.[2] It goes on to assert that "[t]his is far worse precision than the widely used IEEE 754 single precision floating point standard." *Id.* at 2:31-33. Analyzing the claims thus requires some background on how computers handle arithmetic

[2] Citations throughout the draft to the common specification of the patents-in-suit are made to the specification of the '273 Patent.

operations, including the use of floating-point numbers. This background can be gleaned from case law, the common specification, and the IEEE 754 standard that the specification references.

A. Computing processors have long used floating-point number formats.

Processors have for decades performed arithmetic operations using a floating-point number format. Several courts, including the Federal Circuit, have explained this format:

> In floating point format, data is represented by the product of a fraction, or mantissa, and a number raised to an exponent. For example, a number *n* can be represented in base 10 by
>
> $n = m \times 10^e$,
>
> where *m* is the mantissa and *e* is the exponent. If *m* equals 2 and *e* equals 1, *n* equals 20; if *m* equals 2 and e equals -1, then *n* equals 0.2. . . .

Silicon Graphics, Inc. v. ATI Techs., Inc., 607 F.3d 784, 787 (Fed. Cir. 2010); *see also Uniloc USA, Inc. v. Rackspace Hosting, Inc.*, 18 F. Supp. 3d 831, 834 (E.D. Tex. 2013), *Allen v. United States*, 588 F. Supp. 247, 261-63 (D. Utah 1984). The advantage of this floating-point representation is that it widens the range of numbers that can be represented using the same digits, and thus the same number of bits, in a computing representation. In the simplest example, if an integer (whole number) format were used, then two digits would allow representing numbers from 1 to 99. By contrast, in the simplest example of the notation described in *Silicon Graphics* with two digits in base 10, *i.e.*, m and e being allowed one digit each, the range of numbers represented would be from 1 (1×10^0) to 9 billion (9×10^9). With this increased *range* comes reduced *precision*, because while the number 9 or 10 can be expressed exactly (9×10^0 or 1×10^1), there is no precise way to express the number 11, or indeed any number between 8 billion and 9 billion. The ’273 Patent itself quotes a prior art source as explaining how floating-point numbers increase “dynamic range” relative to integer numbers. ’273 Patent at 5:25-30. (This is discussed in more detail below in Part II.C.)

B. The IEEE 754 standard, which the patent specification references, describes a floating-point number's elements and the potential tradeoff between precision and dynamic range.

The IEEE Standard for Floating-Point Arithmetic (IEEE 754), first issued in 1985,[3] established the key aspects of the standardized floating-point number format, including several that provide background for the patents in suit: format and precision.

Format: A binary floating-point number, as specified in the 1985 IEEE 754 standard, has a "numerical value, if any, [that] is [i] the signed product of its [mantissa][4] and [ii] two raised to the power of the exponent." IEEE 754 at 2. A floating-point number must have a finite number of digits in the mantissa for computing; computers cannot encode an infinite number of digits. As the patent specification explains, floating-point numbers allow a much wider range of calculations than integers (*i.e.*, 1, 2, 3, etc.), because a sequence of the same number of digits can represent a much wider range of numbers. '273 Patent at 5:25-30. That is, a floating-point number, like scientific notation, has an exponent that establishes the range of numbers the floating-point format can represent. IEEE 754 at 2. But that additional range comes at some loss of precision: floating-point representations have to be rounded to fit within the number of digits available for the mantissa. *Id.* at 5. By contrast, real numbers can have many or even an infinite number of digits in the mantissa: two examples are (i) the fraction ⅓, which is 0.33 repeating infinitely when represented as a decimal and (ii) the concept of *pi*, which continues infinitely.

Precision. All floating-point number formats must have a specified precision level, which is nothing more than the number of digits in the mantissa. *Id.* at 2 (defining "significand"). The more digits, the more *precise* the numbers the floating-point format can represent. A common

[3] Google is filing a concurrent motion for judicial notice for the 1985 version of the IEEE 754.
[4] IEEE 754 uses the term "significand" to refer to what the case law and other sources call the "mantissa." For simplicity, this Memorandum consistently uses the term "mantissa."

operation used with *pi* demonstrates this. To calculate the circumference of a circle, one multiplies *pi* times the diameter of the circle. So, if the circle has a diameter of exactly 2 centimeters, one would determine the circumference by multiplying *pi* by 2. Using a decimal (*i.e.*, base 10) floating-point format with a 3-digit mantissa, *pi* would be represented as 3.14 x 10^0. Thus, that operation would yield a circumference of 6.28. By contrast, if the format had a 5-digit mantissa, *pi* would be represented as 3.1416 x 10^0. There, the operation would yield a circumference of 6.2832, which is a more precise result.

In computing, the bits (or digits) available for the floating-point number representation in computing are allocated to either the mantissa, the exponent, or some other component (*e.g.*, a sign bit or "not a number" indicator). More digits for the mantissa provide additional precision while more digits for the exponent add range. As reflected in Table 1, the IEEE standard had four different precision levels depending on the number of available digits. IEEE 754 at 3 & Table 1. As Table 1 depicts, the standard also includes "extended" formats, which can have a variable number of digits in the mantissa (a.k.a "significand" or "p" in Table 1) and a wider (and also variable) exponent range, relative to the corresponding "basic" format. Table 1 illustrates that increasing the number of mantissa digits for a "single extended" format increases precision; conversely, increasing the exponent digits increases the dynamic range.

C. The patent specification describes pre-existing implementations that (i) expanded dynamic range by using floating-point number representations and (ii) performed arithmetic processing at multiple precision levels.

Dynamic range refers to the range of values that a number format can reflect. '273 Patent at 2:35-39 ("[T]he processing elements have 'high dynamic range' in the sense that they are capable of operating on inputs and/or producing outputs spanning a range at least as large as from one millionth to one million."). The patent specification, expressly relying on a "late 2008" Wikipedia entry, refers to then-existing implementations in describing both low precision and

high dynamic range as characteristics of the floating-point format that a processor uses. It describes dynamic range with the example that floating-point formats have a broader dynamic range than integers, and it describes lower precision formats as having the advantage of requiring less storage and bandwidth. '273 Patent at 5:11-30.

The specification further discloses that graphics processing units (GPUs)[5] had, at the time of the invention, implementations with 16-bit floating point support "alongside" support for 32-bit floating point and "increasingly, 64-bit floating point." '273 Patent at 5:31-33.

III. CLAIMS

The Complaint alleges infringement of several claims from each of the patents in suit.[6] All the claims have a similar structure and share core claim elements. Accordingly, claim 1 of the '273 Patent ('273 Patent at 29:65-30:16) is used to set out below the elements that are common to all claims alleged to be infringed. (Although claim 1 is not identified in the Complaint as an infringed claim, its dependent claim 17 is one of the identified claims): (a) "a low precision, high dynamic range (LPHDR) execution unit"; (b) "a first input signal representing a first numerical value"; (c) "a first operation" (performed on the first input signal); (d) "a first output signal representing a second numerical value"; (e) "a dynamic range of possible valid inputs to the first operation" where the range, varying by claim, is a numerical span; and (f) "for at least X=[x]% of the possible valid inputs to the first operation, the statistical mean, over repeated execution of the first operation on each specific input from the at least X% of the possible valid inputs to the first operation, of the numerical values represented by the first

[5] GPUs "are a variety of parallel processor that evolved to provide high speed graphics capabilities to personal computers." '273 Patent at 4:64-66.

[6] This Memorandum uses the phrase "asserted claims" to refer collectively to those claims that Singular alleges in its Complaint that Google infringes. Dkt. 1, Compl. ¶ 36 ('273 Patent, claims 17, 18, 52, 53); *id.* ¶ 53 ('156 Patent claims 6, 7, 21, 22); *id.* ¶ 71 ('961 Patent, claims 1-5, 10, 13-15).

output signal of the LPHDR unit executing the first operation on that input differs by at least Y=[y]% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input," where each of x and y has one of two different values depending on the claim.

Elements a-d claim a low-precision, high dynamic range unit to execute computing operations, an input to that unit, an operation on that input, and an output of that operation, where both the input and the output represent numerical values.

Element e claims a span of "possible valid inputs" comprising all the possible first inputs to the first operation. The claims have one of two specific numerical spans in element e, either 1/65,000 through 65,000 or 1/1,000,000 through 1,000,000.[7]

Element f claims a minimum imprecision level for the processor. It does so by requiring two things. First, it specifies a minimum percentage of available inputs from the number span (the "possible valid inputs") on which the operation has to be performed when determining the processor's imprecision level. That is the "X=[x]%" component.[8] Second, it specifies the minimum amount, on average, by which the outputs from doing the first operation using the processor have to differ from the outputs from doing the first operation on the "numerical values," with the claim's underlying assumption being that performing the operation on the "numerical values" would have a more precise result. This difference is the "Y=[y]%" component.[9]

---

[7] Claims 17 and 52 of the '273 Patent; 1-3, 5, 10, 14-15 of the '961 Patent; and 6 and 21 of the '156 Patent include the range 1/65,000 to 65,000. Claims 18 and 53 of the '273 Patent; 4 and 13 of the '961 Patent; and 7 and 22 of the '156 Patent include the range 1/1,000,000 to 1,000,000.
[8] Claims 17 and 52 of the '273 Patent; 6 and 21 of the '156 Patent; and 1-5, 10, 13-15 of the '961 Patent claim x as 10%. Claims 18 and 53 of the '273 Patent and 7 and 22 of the '156 Patent claim x as 5%.
[9] Claims 18 and 53 of the '273 Patent, claims 7 and 22 of the '156 Patent claim y as 0.05%;

Some claims either include an additional element in the applicable independent claim or, as dependent claims, add an element. Claims 21 and 22 of the ’156 Patent and claims 10 and 13-15 of the ’961 patent require a “device” comprising “a plurality of components”; claims 6 and 7 of the ’156 Patent and claims 1-5 of the ’961 Patent add an element of “at least one first computing device adapted to control the operation of the at least one first LPHDR execution unit”; claims 17, 18, 52, and 53 of the ’273 Patent, claims 6, 7, 21, and 22 of the ’156 Patent, and claim 3 of the ’961 Patent specifically require that, in the claimed device, “the number of LPHDR execution units exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating-point numbers that are at least 32 bits wide”; claims 6 and 7 of the ’156 Patent and claims 2 and 3 of the ’961 Patent require one of the following comprised in the computing device: “a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FGPA), a microcode-based processor, a hardware sequencer, and a state machine”; claims 5 and 14 of the ’961 Patent require that the operation—the “first operation”—is a “multiplication operation”; and claim 15 of the ’961 Patent and claims 21 and 22 of the ’156 Patent require that the plurality of components are “bonded” or arranged in a “stack.”

IV. LEGAL STANDARD

The Supreme Court addressed patentability of claims directed to computing articles or methods that performed mathematical operations starting in the 1970s. In *Parker v. Flook*, the Court held that “’if a claim is directed essentially to a method of calculating, using a mathematical formula, even if the solution is for a specific purpose, the claimed method is

---

claims 17 and 52 of the ’273 Patent, claims 6 and 21 of the ’156 Patent, and claims 1-5, 10, 13-15 of the ’961 Patent claim y as 0.2%.

nonstatutory [subject matter under § 101].'" 437 U.S. 584, 595 (1978) (quoting *In re Richman*, 563 F.2d 1026, 1030 (C.C.P.A. 1977)). Prior to that, in *Gottschalk v. Benson*, the Court held that an algorithm for converting "binary coded decimal" (BCD) numbers to "pure" binary numbers lacked patentability because "in practical effect" allowing that patent would have meant allowing a "patent [on] an idea." 409 U.S. 63, 71-72 (1972).

In *Alice Corp. v. CLS Bank Int'l*, the Supreme Court set up a two-stage framework under which a claim falls outside § 101 and is thus not patentable where (1) it is directed to a patent-ineligible concept, *i.e.*, a law of nature, natural phenomenon, or abstract idea, and (2) the particular elements of the claim, considered both individually and as an ordered combination, lack an inventive concept sufficient to transform the nature of the claim' into a patent eligible application. 573 U.S. 208, 217-18 (2014); *Elec. Power Grp. LLC v. Alstom SA*, 830 F.3d 1350, 1353 (Fed. Cir. 2016). The Federal Circuit has "described the first-stage inquiry as looking at the 'focus' of the claims, their 'character as a whole,' and the second-stage inquiry (where reached) as looking more precisely at what the claim elements add." *Elec. Power Grp.*, 830 F.3d at 1353. Furthermore, as part of the overall patentability analysis, the Federal Circuit has distinguished between non-patentable "uses of existing computers as tools in aid of processes focused on 'abstract ideas'" and patentable "computer-functionality improvements." *Id.* at 1354 (distinguishing *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327 (Fed. Cir. 2016)).

V. ARGUMENT

The asserted claims are abstract under the fundamental and still-operative principles set forth in *Gottschalk* and *Flook* in the 1970s, when the Supreme Court first took up questions of the patentability, when applied in computing, of long-standing mathematical principles. The claims do nothing more than take concepts that are inherent in numeric representations—precision and range—and purport to apply them on a computer processor. This is no less abstract

than using a computer to convert from binary-coded decimals to "true" binary as in *Gottschalk* or to run a known algorithm for calculating alarm limits for an industrial process as in *Flook*.

Applying *Alice*'s two-part patentability rubric confirms that the asserted claims are not directed to patentable subject matter. Under *Alice*'s first step, the asserted claims are directed to the abstract idea of using a LPHDR execution unit to execute an operation on a first signal representing a numerical value to produce a second numerical value that is intentionally imprecise. The notion of doing imprecise calculations is part of human history, including the ancient Greeks calculating a square root through iterative approximation[10] and a modern-day diner trying to approximate out a 20% tip quickly in a restaurant. The asserted claims rest on the same idea: don't waste time or resources on needless precision when close enough will suffice. Doing such low-precision calculations on numbers that have a high dynamic range is equally abstract because a high dynamic range is just a span of values that a numerical representation can encompass. To take an example from long-extant principles outside computing, a single-digit representation using the Arabic numeral system can represent 9 different values, which is not an especially high dynamic range. By contrast, a two-digit representation for base 10 scientific notation can represent, as set forth above, values ranging from 1 to 9 billion. High dynamic range is, in the literal sense, just mathematics, and is completely abstract, whether used with low-precision calculations or higher-precision calculations.

Nor does the second step under *Alice* salvage the claims' patentability. Rather than evidence some "inventive step," the asserted claims simply apply well-known mathematical ideas using existing computing technology. The claims are directed to LPHDR processing. But

[10] *See, e.g.*, Michael Lahanas, *Heron's Mathematics*, http://www.hellenicaworld.com/Greece/Science/en/HeronsMath.html.

the specification admits that existing implementations of computer processors intentionally used lower precision arithmetic. The specification also admits that existing computer processors used the floating-point number format to increase dynamic range; besides, dynamic range is just a span of numbers and is an inherent part of any number representation. And although the claims purport to specify a *minimum* imprecision level for the claimed processing, the specification does not identify the benefits of any particular level of imprecision for any specific application. Similarly, while the claims also specify particular dynamic ranges, the specification identifies no benefits of any particular dynamic range for any specific application. The asserted claims are also not directed to, nor does the specification identify, any improvement to computing technology, such as hardware or software for performing LPHDR calculations. Instead, the specification just describes the known fact that some applications may require less precision than others, and that giving up a degree of precision may allow a broader range. There is therefore no inventive concept that would salvage patentability at *Alice* step two.

A. The asserted claims are directed to the abstract idea of doing an arithmetic calculation with an intended degree of imprecision to achieve a result more quickly or with fewer resources.

The language of each asserted claim confirms that they are directed to the abstract idea of performing low-precision arithmetic using numbers with relatively high dynamic range. Specifically, the asserted claims are directed to the abstract idea of using (a) a "low precision high dynamic range (LPHDR) execution unit" to (b) "execute a first operation on a first input signal representing a first numerical value" to (c) "produce" a "second numerical value." Looking at the focus of the claims and their character as a whole, they are abstract under *Alice* step one.

The term "low precision" means that the execution unit is imprecise, yielding a "second numerical value" that varies by more than a threshold amount from what the value would have

been had the claimed "exact mathematical calculation of the first operation" been performed at a higher level of precision than what the LPHDR execution unit performs and outputs.[11] As the specification explains, instead of using computing power to achieve the precisely correct mathematical result, it is sometimes preferred to have an anticipated range of imprecision that is acceptable to whatever application that calculation is being applied. '273 Patent at 7:27-39. This is the same (abstract) idea as calculating a tip on a dinner check of $24.76 as $5 rather than trying to calculate the exact value of 20% of the dinner check: you know that $5 is 20% of $25 and thus a much easier number to calculate and that it will differ from the exact value by less than 20 cents, *i.e.*, 20% of $1. Most diners are willing to pay a few cents over 20% in order to save the time and effort of exactly calculating the tip.

Executing a low-precision calculation in a unit that has "high dynamic range" means only that the execution unit can represent a relatively broad range of numbers. This includes small numbers such as 1/65,000 as well as large numbers such as 65,000. '273 Patent at 29:65-30:16. Wide dynamic ranges have long existed outside computing. In the instance of tipping described above, a diner may use approximate numbers to calculate the tip more easily whether it's a $9.76 check at a diner for one person, $24.76 for multiple people, or $148.57 at a gourmet restaurant. In the realm of mathematics specifically, as school children who have learned scientific notation know, the more digits one uses for the exponent, the larger the range of values that the notation format can cover. In fact, any numeric representation, if it has a limited number of digits, will have a range of values it covers. The patent specification itself recognizes this abstract idea as

[11] The claims are unspecific about what the lower and higher levels of precision are and about how the precision is lower; but they purport to claim a lower precision output than what otherwise might be achieved by performing the "first operation" on a "numerical value" rather than on a "first input."

present in pre-existing implementations, noting that a floating-point representation offers wider dynamic range than integers. '273 Patent at 5:25-30.

The claimed low-precision arithmetic and high-dynamic range execution unit is entirely conventional and does not constitute an improvement to computing technology under either *Enfish* or *DDR Holdings v. Hotels.com, LP*, 773 F.3d 1245, 1259 (Fed. Cir. 2014). Indeed, the specification itself identifies processor implementations that intentionally used low precision and also had high dynamic range. '273 Patent at 5:11-30; *see infra* Part V.B.1.–2. (explaining why claims lack an inventive concept).

To the extent there is any suggestion of an invention in the specification, it appears, at most, to be the realization that LPHDR processing may be useful in certain applications, '273 Patent at 7:5-11; however, that "realization" is found nowhere in the claims, which attempt only to claim a *result* rather than a particular inventive application of the abstract idea. *See Internet Patents Corp. v. Active Network, Inc.*, 790 F.3d 1343, 1348 (Fed. Cir. 2015); *see also Am. Axle & Mfg. Co. v. Neapco Holdings LLC*, 939 F.3d 1355, 1363 (Fed. Cir. 2019) ("[F]eatures that are not claimed are irrelevant as to step 1 or step 2 of the *Mayo/Alice* analysis."). In *Internet Patents Corp.*, the Federal Circuit rejected a claim that "contains no restriction on how the result is accomplished." 790 F.3d at 1348. Similarly, in *Electric Power Group*, the Federal Circuit explained that there exists "an important common-sense distinction between ends sought and particular means of achieving them, between desired results (functions) and particular ways of achieving (performing) them." 830 F.3d at 1356. For that reason, "the essentially result-focused, functional character of claim language has been a frequent feature for claims held ineligible under § 101 . . . ." *Am. Axle & Mfg.*, 939 F.3d at 1365 (the "distinction between results and means is fundamental to the step 1 eligibility analysis . . . ."). *Id.* Singular's attempt to claim the

abstract idea of performing LPHDR calculations "without specifying the means of how to implement the concept is ineligible under section 101." *Id.*

In sum, all the Asserted Claims are directed to an abstract idea, thus requiring consideration of the *Alice* step-two inquiry to determine patentability.

B. The limitations of the asserted claims do not add any inventive concept that transforms the abstract idea into a patentable one.

Under *Alice* step two, the question is whether a claim nonetheless "contains an inventive concept sufficient to transform the claimed abstract idea into a patent-eligible application." 573 U.S. at 221 (citations omitted). At this second step, the focus is on the individual elements or the ordered combination *apart from the abstract idea. Chamberlain Grp., Inc. v. Tectronic Indus., Co.*, 935 F.3d 1341, 1349 (Fed. Cir. 2019) (emphasis added). Thus, "a claimed invention's use of the ineligible concept to which it is directed cannot supply the inventive concept that renders the invention 'significantly more' than that ineligible concept.'" *BSG Tech LLC v. BuySeasons, Inc.*, 899 F.3d 1281, 1290 (Fed. Cir. 2018). Here, no elements in the asserted claims provide an inventive concept sufficient to establish patentability.

1. None of the core elements of the asserted claims adds an inventive concept to the abstract idea.

First, element a (*i.e.*, low-precision, high dynamic range processor) is not inventive. The element offers no detail or limitation on the processing apparatus itself, other than that it is a conventional low-precision and high dynamic range processor. Indeed, claiming a LPHDR processor is nothing more than the abstract idea itself: doing low-precision arithmetic using inputs with a high dynamic range. *Cf. Chamberlain*, 935 F.3d at 1349 (inventive concept or ordered combination must go beyond the abstract idea itself); *BSG Tech LLC*, 899 F.3d at 1290 (same). As discussed above, that idea is itself entirely conventional, both in daily life and computing. In daily life, people often use low-precision arithmetic when an exact result is not

necessary to accomplish the purpose of the calculation, whether it be calculating a tip or splitting a taxicab fare. In the field of computing, the specification admits that low-precision arithmetic was also conventional: in the discussion of GPUs referenced in Part II.C, the specification states that then-existing processor implementations used 16-bit representations when the application in question did not require the more standard, and higher-precision, 32-bit representation. '273 Patent at 5:11-35. The 1985 IEEE 754 standard that the specification references says that loss of precision is a concept inherent in a floating-point number representation, because "[n]ormally, a result is rounded to the precision of its destination." IEEE 754 at 6.

The specification also confirms the conventionality of high-dynamic range in computing and discusses existing approaches for representing a high dynamic range. '273 Patent at 5:11-25. It goes on to confirm then-existing implementations expanded dynamic range by using floating-point representations rather than integers. *Id.* at 5:25-30. The referenced IEEE 754 standard provided for an extended format with variable range depending on the number of exponent digits used. IEEE 754 at 3. Outside the field of computing, scientific notation has embodied the idea that using more digits for an exponent widens the range of values that a particular notation form can cover, and that using more digits for the mantissa will increase precision.

Nor is there anything inventive about a processor that performs low precision arithmetic on number formats with high dynamic range. Levels of precision and a range of values exist in any numerical representation, including floating-point numbers, and computer processors have conventionally used floating-point numbers for decades. The IEEE 754 standardization of floating-point representations, which was for the purpose of computer processing, took place in 1985. Any representation compliant with that standard will have some degree of precision and some degree of dynamic range. The fact that the patent specification purports to claim a high

dynamic range is not patentable, just as claiming the monitoring and calculation of conventional alarm limits using a computer was not patentable in *Flook.* 437 U.S. at 594-95, 596-98.

Given the conventionality of low-precision arithmetic and high dynamic range in both computing and non-computing fields, "the focus of the claims" in this instance "is not on such an improvement in computers as tools, but on certain independently abstract ideas"—use of low-precision, high-dynamic range arithmetic—"that use computers as tools." *Elec. Power Grp.*, 830 F.3d at 1354. Furthermore, just "limiting the claims to the particular technological environment" of low-precision, high dynamic range processors "is, without more, insufficient to transform them into patent-eligible applications of the abstract idea at their core." *Id*. In *Flook*, the Supreme Court noted that "respondent's claim is, in effect, comparable to a claim that the formula $2\pi r$ can be usefully applied *in determining the circumference of a wheel.*" 437 U.S. at 595 (emphasis added). In other words, taking a known, abstract idea about how to determine the circumference of a circle does not become patentable simply by applying it to the technical field of wheels. Here, applying the known, abstract idea of low-precision arithmetic across a high-dynamic range of numbers doesn't become patentable simply because it is performed on a computer processor.

Second, elements b, c, d (*i.e.*, first input signal, first operation, and first output signal) do no more than claim, generically, that a mathematical operation is to be performed; they do not claim anything beyond the abstract idea and are entirely conventional. "Respondent's process is unpatentable under § 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention." *Flook*, 437 U.S. at 594.

Third, the recitation of a minimum (dynamic) range for the representation of the numbers is not inventive. Instead, a minimum range simply amounts to a minimum number of bits to use

for the exponent of a computer's representation format. "[M]ere field-of-use limitations are generally insufficient to render an otherwise ineligible process claim patent-eligible." *In re Bilski*, 545 F.3d 943, 957 (Fed. Cir. 2008) (citing *Diamond v. Diehr*, 450 U.S. 175, 191-92 (1981)), *aff'd sub nom. Bilski v. Kappos*, 561 U.S. 593 (2010). The idea that using additional exponent bits would expand dynamic range is inherent in scientific notation and was included in the 1985 version of the IEEE 754 standard in the "extended" formats. IEEE 754 at 3 and Table 1. Even if the claimed ranges purported to improve on the IEEE 754 standard for floating-point number representations, that alone would not render them inventive. *See Uniloc USA, Inc.*, 18 F. Supp. at 838 (holding claim unpatentable under Section 101 where it "merely constitutes an improvement on the known method for processing floating-point numbers"). Regardless, the claims are much like the application the Supreme Court found unpatentable in *Flook*, because there is no purported invention or benefit described for the particular imprecision levels or dynamic ranges in the claims. *Compare* '273 Patent at 27:5-7 ("Besides having various possible degrees of precision, implementations may vary in the dynamic range of the space of values they process.") *with Flook*, 437 U.S. at 586 ("The patent application does not purport to explain how to select the appropriate margin of safety, the weighting factor, or any of the other variables.").

American Axle & Mfg. Co. v. Neapco Holdings LLC found a claim unpatentable in similar circumstances. 309 F. Supp. 3d 218, 225 (D. Del. 2018), *aff'd* 939 F.3d 1355 (Fed. Cir. 2019). There, the claim at issue applied to automotive parts a natural law of physics regarding the force required to extend or compress a spring (Hooke's law) and where the claim included specific numerical parameters regarding how the principle would be applied. *Id.* As the court explained, "the Asserted Claims do not disclose a method of manufacturing a propshaft," which was the automotive part at issue, just as the asserted claims here are not directed to any novel

way of manufacturing a LPHDR processor. "[I]nstead, considered as a whole, [the claims] are directed to the mere application of Hooke's law, and they fail to instruct ***how*** to design the tuned liners or manufacture the driveline system" (parts of the propshaft) "to attenuate vibrations" (the purported improvement). *Id.* (emphasis in original). The same holds true here: the asserted claims do not recite any particular means of HDR representation; nor do they (or the specification) tie the ranges included in the claim to improved utility of HDR in any particular application. Rather, the asserted claims simply claim the abstract idea of processing low-precision arithmetic with high dynamic range.

Fourth, the recitation in the claims of a particular threshold for the imprecision of the execution unit is not inventive. Notably, the claims require only that the calculated answers *exceed* a specified level of error, and thus would be satisfied even if the execution unit reported the same number (*e.g.*, zero) for every input. Moreover, the specification disclaims the significance of any particular level of imprecision, noting that the mean error rate, *i.e.*, "y" in element f of the claims, could be more than 0.05%, 0.1%, 0.2%, 0.5%, 1%, 2%, 5%, 10% or 20%. '273 Patent at 27:58-60. The particular fraction of the results for which that mean error applies likewise is disclaimed as having any significance, with examples cited including 1%, 2%, 5%, 10%, 20% or 50%. *Id.* at 27:54-56. Indeed, even those many options for imprecision levels and fractions of the results to which they apply are "merely examples" that "do not constitute limitations of the present invention." *Id.* at 27:60-62. As in *Electric Power Group* and *American Axle*, this element of the claim does nothing more than instruct applying the abstract idea of an imprecise mathematical calculation to achieve a desired result, without offering "a particular concrete solution to a problem." Accordingly, it does nothing more than "attempt[] to patent the abstract idea of a solution to a problem in general." *Elec. Power Grp.*, 830 F.3d at 1356.

Thus, none of the core elements in the asserted claims, *i.e.*, LPHDR processing with certain dynamic ranges and acceptable imprecision levels specified, provide an inventive concept at step two of the *Alice* inquiry.

2. None of the elements that various claims add to the core elements add an inventive concept.

For those claim elements that merely specify a device with a "plurality of components," this, at most, states that the abstract idea is applied on a computer. That does not add any inventive concept. *Alice*, 573 U.S. at 222. The same is also true for those claims including the element of a computer adapted to control the at least one first LPHDR execution device. *See* '273 Patent at 4:64-66 (noting that specialized graphics processing units may be included in personal computers). This also holds true for those claim elements that require the plurality of components to be "bonded" or "stacked." The specification discloses using conventional techniques for accomplishing this bonding or stacking. '273 Patent at 26:5-16. The same is true for those elements that specify a central processing unit (CPU), a graphics processing unit (GPU), a field programmable gate array (FGPA), a microcode-based processor, a hardware sequencer, and a state machine. These are standard, preexisting components that the patent does not purport to improve. *Id.* at 3:30-39 (stating that these various types of processors existing at the time of the specification).

For those claim elements that specify the number of LPHDR units must exceed by at least 100 the number of 32-bit wide execution units, the number 100 is like the claimed dynamic ranges and error rates—that is, the patents offer no reason why this quantity reflects a definite solution and identify no application where that particular threshold difference between LPHDR and non-LPHDR units would offer a benefit. Furthermore, the specification itself discloses preexisting implementations of computers with array processors, *e.g.*, processors with many

elements or engines to performing arithmetic, including so-called "SIMD" machines and GPUs, which the specification describes as having multiple arithmetic elements. '273 Patent at 3:40-56 and 5:1-10. As to GPUs, the specification says that their collection of arithmetic elements may include both LPHDR and non-LPHDR processing: "When a graphics processor includes support for 16 bit [lower precision] floating point, that support is *alongside* support for 32 bit [higher precision] floating point, and increasingly, 64 bit [higher precision] floating point." '273 Patent at 5:31-33 (emphasis added). Notably, the specification describes both GPUs and, in at least one potential implementation, LPHDR processors, as having large numbers of parallel units to perform arithmetic operations. *Compare* '273 Patent at 4:64-5:6 *with id.* at 6:51-63.

For those claim elements requiring the first operation to be a multiplication operation, that too is not inventive. Multiplication as an arithmetic operation has existed since time immemorial, and the specification does not identify any inventive concept in limiting the first operation to multiplication. Rather, this is just an aspect of floating-point arithmetic, which is not patentable in and of itself. *See Uniloc USA, Inc.*, 18 F. Supp. 3d at 838 (claim on rounding floating-point number prior to, rather than after, arithmetic operation not patentable).[12]

VI. CONCLUSION

For the foregoing reasons, all the claims identified in Plaintiffs' Complaint as potentially infringed should be found unpatentable under Section 101, and this action should be dismissed.

[12]There is nothing about the ordered combination of elements that reflects an inventive concept. The limited additional claim elements discussed in this Part V.B.2 are not described in the specification as adding any inventive concept by ordering the processor's operations in a particular way; instead, they are just describing the result of applying the abstract idea using known elements of computing technology. *See Maxon, LLC v. Funai Corp., Inc.*, 255 F. Supp. 3d 711, 720 (N.D. Ill. 2017), *aff'd* 726 Fed. App'x 797 (2018) (describing result that uses generic computing components insufficient to constitute inventive ordered combination).

Date: February 28, 2020

Respectfully submitted,

/s/ Gregory F. Corbett
Gregory F. Corbett (BBO #646394)
gregory.corbett@wolfgreenfield.com
Nathan R. Speed (BBO # 670249)
nathan.speed@wolfgreenfield.com
Elizabeth A. DiMarco (BBO #681921)
elizabeth.dimarco@wolfgreenfield.com
WOLF, GREENFIELD & SACKS, P.C.
600 Atlantic Avenue
Boston, MA 02210
Telephone: (617) 646-8000
Fax: (617) 646-8646

Robert Van Nest*
rvannest@keker.com
Matthias Kamber*
mkamber@keker.com
Michelle Ybarra*
mkamber@keker.com
Jay Rapaport*
jrapaport@keker.com
Andrew Bruns*
abruns@keker.com
Deeva Shah*
dshah@keker.com
KEKER, VAN NEST & PETERS LLP
633 Battery Street
San Francisco, CA 94111-1809
(415) 391-5400

Michael S. Kwun *(pro hac vice pending)*
mkwun@kblfirm.com
Asim Bhansali *(pro hac vice pending)*
abhansali@kblfirm.com
KWUN BHANSALI LAZARUS LLP
555 Montgomery Street, Suite 750
San Francisco, CA 94111
(415) 630-2350

Counsel for Defendant Google LLC
**motions for pro hac vice to be filed*

CERTIFICATE OF SERVICE

I certify that this document is being filed through the Court's electronic filing system, which serves counsel for other parties who are registered participants as identified on the Notice of Electronic Filing (NEF). Any counsel for other parties who are not registered participants are being served by first class mail on the date of electronic filing.

Date: February 28, 2020

/s/ Gregory F. Corbett
Gregory F. Corbett